

(19) KOREAN INTELLECTUAL PROPERTY OFFICE

KOREAN PATENT ABSTRACTS

(11)Publication number: **1019990082741 A**
(43)Date of publication of application:
25.11.1999

(21)Application number: **1019990008628**
(22)Date of filing: **15.03.1999**
(30)Priority: **20.04.1998 1**

(71)Applicant: **INTERNATIONAL BUSINESS
MACHINES CORPORATION**
(72)Inventor: **MATTHEWS GARETH
CHRISTOPHER
MEDINA DAVID
WYNN ALLEN CHESTER**

(51)Int. Cl. **G06F 15/16**

(54) **APPARATUS AND A METHOD FOR HIBERNATION IN A DISTRIBUTED DATA PROCESSING SYSTEM,
ESPECIALLY FOR MANAGING AND BOOTING THE DATA PROCESSING SYSTEM**

(57) Abstract:

PURPOSE: An apparatus and a method for hibernation in a distributed data processing system are provided to acquire an improved method and apparatus for managing the data processing system and booting the data processing system, thereby improving the convenience for a user. CONSTITUTION: A method for hibernation in a distributed data processing system comprises the following steps of: receiving data used for initializing the data processing system in the data processing system from a remote network source(300); saving an image(314) of the data processing system as a selected state in the data processing system after initializing the data processing system; and recovering the data processing system to the selected state using the saved image.

copyright KIPO 2007

Legal Status

Date of request for an examination (19990824)
Notification date of refusal decision (00000000)
Final disposal of an application (registration)
Date of final disposal of an application (20011206)
Patent registration number (1003189750000)
Date of registration (20011214)

Number of opposition against the grant of a patent ()
Date of opposition against the grant of a patent (00000000)
Number of trial against decision to refuse ()
Date of requesting trial against decision to refuse ()
Date of extinction of right ()

(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) 。 Int. Cl. ⁶
G06F 15/16

(45) 공고일자 2002년01월09일
(11) 등록번호 10- 0318975
(24) 등록일자 2001년12월14일

(21) 출원번호 10- 1999- 0008628
(22) 출원일자 1999년03월15일

(65) 공개번호 특1999- 0082741
(43) 공개일자 1999년11월25일

(30) 우선권주장 09/062,885 1998년04월20일 미국(US)

(73) 특허권자 인터내셔널 비지네스 머신즈 코포레이션
포만 제프리 엘
미국 10504 뉴욕주 아몬크

(72) 발명자 매튜스가레스크리스토퍼
미국텍사스78613세다르파크맥카우드라이브2303
메디나데이비드
미국텍사스78717오스틴카프리아일LN.17030
원알렌체스터
미국텍사스78664라운드락이글스네스트스트리트3816

(74) 대리인 김성택
허정훈

참사관 : 김준학

(54) 분산 데이터 프로세싱 시스템 내에서 하이버네이션을 위한 방법 및 장치

요약

본 발명은, 분산 데이터 프로세싱 시스템(distributed data processing system) 내에서 데이터 프로세싱 시스템을 초기화하기 위한 방법 및 장치에 관한 것이다. 데이터 프로세싱 시스템은 데이터를 수신한다. 여기서 데이터는 데이터 프로세싱 시스템을 초기화하기 위해 사용된다. 데이터 프로세싱 시스템은, 데이터 프로세싱 시스템의 초기화 이후에 선택된 상태로 데이터 프로세싱 시스템 내에 이미지(image)를 저장(save) 한다. 여기서 이미지는 저장된 이미지이다. 데이터 프로세싱 시스템은 저장된 이미지를 사용하여 그 상태로 회복한다.

대표도
도 3

색인어

분산 데이터 프로세싱시스템, 하이버네이션

명세서

도면의 간단한 설명

도 1은 본 발명의 바람직한 실시예에 따른 분산 데이터 프로세싱 시스템(distributed data processing system)의 다이어그램.

도 2는 본 발명이 실행될 수 있는 데이터 프로세싱 시스템의 블록 다이어그램.

도 3은 본 발명의 바람직한 실시예에 따른 네트워크 컴퓨터(network computer)를 관리하는데 사용되는 컴포넌트(component)의 블록 다이어그램.

도 4는 본 발명의 바람직한 실시예에 따른 이미지(image)의 다이어그램.

도 5는 본 발명의 바람직한 실시예에 따른 하이버네이션(hibernation)을 위한 네트워크 컴퓨터를 구성하기 위한 프로세스(process)의 플로우차트.

도 6은 본 발명의 바람직한 실시예에 따른 네트워크 컴퓨터를 부팅(booting)하기 위한 프로세스의 플로우차트.

* 도면의 주요 부분에 대한 부호의 설명

300 서버(server) 302 부트 이미지(boot image)

310 NC(network computer) 316 NC 파일(NC file)

402 헤더(header) 404 상태(state)

406 페이징 파일(paging file)

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 개선된 분산 데이터 프로세싱 시스템(distributed data processing system)에 관한 것으로, 특히 분산 데이터 프로세싱 시스템 내에서 데이터 프로세싱 시스템을 관리하기 위한 개선된 방법 및 장치에 관한 것이다. 더욱 특별하게는, 분산 데이터 프로세싱 시스템 내에서 데이터 프로세싱 시스템을 부팅하기 위한 개선된 방법 및 장치를 제공하는 것에 관한 것이다.

컴퓨터는 물리적 기계, 즉 하드웨어와 물리적 기계를 작동하게 하는 명령, 즉 소프트웨어의 모두를 포함한다. 소프트웨어는 애플리케이션(application) 및 오퍼레이팅 시스템(operating system) 프로그램을 포함한다. 프로그램이 단순히, 특정의 문제를 해결하는 것과 같은, 유저(user)를 위한 작업을 하는 것이라면 그것은 애플리케이션 소프트웨어로 언급될 것이다. 프로그램이 컴퓨터의 하드웨어 및 애플리케이션 프로그램의 실행을 제어한다면, 그것은 오퍼레이팅 시스템 소프트웨어로 불린다. 시스템 소프트웨어는 오퍼레이팅 시스템, 실제 컴퓨터 또는 중앙 처리 유닛(central processing unit: CPU)를 제어하는 프로그램, 그리고 프린터(printer) 및 터미널(terminal)과 같은 입력 및 출력 장치(I/O)

를 제어하는 디바이스 드라이버(device driver)를 더 포함한다.

범용 컴퓨터(general purpose computer)는 매우 복잡하다. 일반적으로 애플리케이션 프로그램의 큐(queue)가 CPU를 사용하기 위해 기다리고 있는 상태이다. 오퍼레이팅 시스템은 어느 프로그램이 다음에 동작할 것인지와 그것에게 얼마만큼의 CPU 시간을 사용하게 할 것인지, 그리고 애플리케이션에게 어떤 다른 컴퓨터 자원을 사용하게 할 것인가를 결정할 필요가 있을 것이다. 더욱이, 각 애플리케이션 프로그램은 특별한 입력 또는 출력 장치를 요구할 것이고, 또한 애플리케이션 프로그램은 그것의 데이터를 디바이스 드라이버를 제어하는 오퍼레이팅 시스템으로 전송하여야 한다.

컴퓨터가 부팅(boot)될 때, ROM(read only memory)에 저장된 부팅 프로그램은 컴퓨터의 메모리 상으로 오퍼레이팅 시스템의 로딩(loading)을 시작하기 위해 사용된다. 용어 '부팅'(boot)은 컴퓨터를 시작하거나 리셋팅(resetting)하는 프로세스(process)를 언급한다. 처음으로 켜지거나(cold boot) 리셋될 때(warm boot), 컴퓨터는, 컴퓨터의 더욱 복잡한 오퍼레이팅 시스템을 로드하고 시작하며 사용하기 위해 그것을 준비하는 소프트웨어를 실행한다. 따라서, 컴퓨터는 그 자신의 부트스트랩(bootstrap)에 의하여 그 자신을 세운다(pull up)라고 말해진다. 부트 프로그램은 컴퓨터 상으로 오퍼레이팅 시스템을 로드하기 위해 사용되는 '부트 블록' 데이터 프로그램('boot block' dataprogram)으로 불리는, 보다 큰 부트 프로그램을 어디에서 찾을 것인가를 컴퓨터에 명령한다. 용어 '부트 블록'('boot block')은 오퍼레이팅 - 시스템 로더(loader) 및 컴퓨터를 시작하게 하는 다른 기본적 정보를 포함하는 디스크 부분을 말한다. 단독 - 운영(stand - alone) 컴퓨터에서, 부트 블록 프로그램 및 오퍼레이팅 시스템은 로컬 하드 드라이브(local hard drive) 상에서 찾을 수 있다.

다수의 컴퓨터를 포함하는 네트워크는, 하나 또는 그 이상의 통신 링크 상에서 서로 통신하는 '노드'('node') 또는 '네트워크 컴퓨터'('network computer')로 지칭되는 컴퓨터를 구비하여 컴퓨터 네트워크의 집합체(aggregation)를 이룸으로써 형성될 수 있다. 오늘날, 많은 컴퓨터 워크스테이션(work station)이 다른 워크스테이션, 파일 서버(file server), 또는 LAN(local area network) 상의 다른 자원으로 연결된다. 네트워크 상의 각 컴퓨터는 네트워크로 통신 링크를 설정할 능력을 제공하는 어댑터 카드(adapter card) 또는 다른 유사한 수단을 통해 네트워크로 연결된다.

네트워크 컴퓨터(network computers: 이하 NCs라 한다)의 관리 측면에서, 다른 NCs 사이에서의 프로그램, 오퍼레이팅 시스템, 그리고 구성(configuration)의 균일성(uniformity)을 유지하는 것이 필요하다. 균일성을 관리함에 있어서, 원격 부트 오퍼레이션을 사용하는 기술이 네트워크 환경에 있는 NCs를 지원하기 위해 사용될 수 있다. 그러한 경우에, 각 네트워크 컴퓨터(NC)는 원격 부트 디스크, 네트워크에 연결된 서버 또는 디스크 어레이(disk array) 시스템과 같은 네트워크 상의 어딘가에 위치한 다른 디바이스로부터 부팅된다. 그러한 부트 시스템은, 시스템 관리자(system administrators)가 각 NC에서 물리적으로 재구성하거나 또는 애플리케이션을 수정할 필요가 없기 때문에, 개별적 NCs를 갱신하는데 필요한 시간을 최소화할 수 있도록 한다. 더욱이, 원격 부트 프로세스는 완전히 디스크 없는 NCs를 지원한다. 또한, 원격 부트 파일이 안전한 위치 안에 보존될 수 있고, 복사본이 네트워크에 있는 NCs 사이에서 분산될 필요가 없기 때문에 원격 부트 프로세스는 소프트웨어 및 네트워크의 보안성을 향상시킨다.

원격 부팅이 가지는 하나의 문제는, 부팅시에 부트 이미지(boot image)가 네트워크 상에서 전송되어야 하고, 또한 제한된 수의 NCs만이 동시에 부팅될 수 있기 때문에 개별 NC를 부팅하는데 필요한 시간이 필요 이상으로 길다는 점이다. 예를 들면, 가령 네트워크 트래픽(network traffic), 이미지 사이즈(image size), 그리고 초기화 시간에 의존하는, NC를 부팅하는데 걸리는 시간은 15 분 또는 그 이상이 소요될 수 있다. 네트워크가 갑자기 다운되고, 그 후 다시 시작되는 경우에 이 문제는 더욱 심각해진다. 네트워크가 중단되고 다시 시작될 때, 모든 NCs가 네트워크로부터 부팅하고 에

플리케이션을 로딩하는 부트 스톰(**boot storm**)이 발생할 수 있다. 모든 NCs가 네트워크로부터 오퍼레이팅 시스템 및 애플리케이션의 다운로드(**download**)를 시도하고 있기 때문에, 이 상황은 심각한 대역폭 문제를 낳을 수 있고 NCs를 부팅하는데 필요한 시간을 크게 증대시킨다.

많은 사무실 환경에서, NCs를 부팅하거나 저장하는데 있어서 그러한 지연은 받아들여질 수 없다. 예를 들면, 은행같은 사무실 환경에서, 거래(**transaction**)가 발생하기 전에 고객으로 하여금 15 분 또는 그 이상을 기다리게 하는 것은 비즈니스(**business**) 실무상 받아들여질 수 없다. 하지만, 고객 거래를 위해 사용되는 NC가 갑자기 다운된 후 리부팅될 때 이 상황이 발생한다.

그러므로, 네트워크에 연결된 네트워크 컴퓨터를 부팅하고 그리고 리부팅(**rebooting**)하기 위한 개선된 방법 및 장치를 가진다면 유익할 것이다.

발명이 이루고자 하는 기술적 과제

본 발명의 목적은 개선된 분산 데이터 프로세싱 시스템(**distributed data processing system**)을 제공함에 있다.

본 발명의 다른 목적은 분산 데이터 프로세싱 시스템 내에서 데이터 프로세싱 시스템을 관리하기 위한 개선된 방법 및 장치를 제공함에 있다.

본 발명의 또 다른 목적은 분산 데이터 프로세싱 시스템 내에서 데이터 프로세싱 시스템을 부팅(**booting**)하기 위한 개선된 방법 및 장치를 제공함에 있다.

발명의 구성 및 작용

본 발명은 분산 데이터 프로세싱 시스템 내의 데이터 프로세싱 시스템을 초기화하기 위한 방법에 관한 것이다. 데이터 프로세싱 시스템은 데이터를

수신한다. 여기서 데이터는 데이터 프로세싱 시스템을 초기화하기 위해 사용된다. 데이터 프로세싱 시스템은, 데이터 프로세싱 시스템의 초기화 이후에 선택된 상태로 데이터 프로세싱 시스템 내에 이미지를 저장한다(**save**). 여기서 이미지는 저장된 이미지이다. 데이터 프로세싱 시스템은 저장된 이미지를 사용하여 그 상태로 회복한다.

본 발명의 특징이라고 믿어지는 신규한 특징은 첨부된 청구항들에 설정되어 있다. 사용상의 바람직한 형태뿐만 아니라, 발명 그 자체, 더욱이 그것의 목적 및 장점이 첨부된 도면과 결합되어 읽혀질 때, 서술적인 실시예의 상세한 설명을 따라 참조함으로써 가장 잘 이해될 것이다.

도 1에 따르면, 분산 데이터 프로세싱 시스템의 다이어그램(**diagram**)이 본 발명의 바람직한 실시예에 따라 도시되어 있다. 분산 데이터 프로세싱 시스템(100)은 본 발명이 실행될 수 있는 컴퓨터들의 네트워크(**network of computers**)이다. 분산 데이터 프로세싱 시스템(100)은 네트워크(102)를 포함하고, 그 네트워크는 분산 데이터 프로세싱 시스템(100) 내에서 서로 연결된 여러 가지 장치 및 컴퓨터 사이에 통신 링크를 제공하기 위해 사용되는 매체이다. 네트워크(102)는 와이어(**wire**) 또는 광섬유 케이블(**fiber optic cable**)과 같은 영구적 연결 또는 전화 연결을 통해 만들어지는 일시적 연결을 포함할 수 있다.

도시된 예에서, 서버(104)는 저장 유닛(106)와 함께 네트워크(102)로 연결된다. 또한, 네트워크 컴퓨터(108, 110 및 112) 자체가 네트워크(102)에 연결된다. 이러한 응용 목적을 위해서 네트워크 컴퓨터는 네트워크에 결합된 다른 컴퓨터로부터 부트 이미지(**boot image**)를 수신하는, 네트워크에 결합된 임의의 컴퓨터이고 또한 서버 관리(**server managed**) 컴퓨터일 수도 있다. 서버(104)는 부트 파일(**boot file**), 오퍼레이팅 시스템 이미지(**operating system im**

age), 그리고 애플리케이션과 같은 데이터를 NCs(108-112)로 제공한다. NCs(108,110 및 112)는 서버(104)에 대해 클라이언트(client)들이다. 분산 데이터 프로세싱 시스템(100)은 도시되지 않은 추가의 서버들, NCs, 그리고 다른 장치들을 포함할 수 있다. 도 1은 예시로서 나타내어지는 것이며, 본 발명의 프로세스에 대한 구조적 제한으로서 나타내어지는 것은 아니다.

도 2에, 본 발명이 실행될 수 있는 데이터 프로세싱 시스템(200)의 블록 다이어그램이 나타나 있다. 데이터 프로세싱 시스템(200)은 PCI(peripheral component interconnect) 로컬 버스 구조(local bus architecture)를 채용한다. 도시된 예가 PCI 버스를 채용하지만, 마이크로 채널(Micro Channel) 및 ISA 같은 다른 버스 구조가 사용될 수 있다. 프로세서(202) 및 메인 메모리(204)는 PCI 브리지(bridge)(208)를 통해 PCI 로컬 버스(206)로 연결된다. PCI 브리지(208)는 또한 프로세서(202)를 위해 집적 메모리 컨트롤러(integrated memory controller) 및 캐쉬 메모리(cache memory)를 포함할 수 있다. PCI 로컬 버스(206)로의 추가적인 연결들이 직접 컴포넌트 상호 연결(direct component interconnection) 또는 애드-인 보드(add-in board)를 통해 구현될 수 있다. 도시된 예에서, LAN 어댑터(local area network adapter)(210), SCSI 호스트 버스 어댑터(host bus adapter)(212), 그리고 확장 버스 인터페이스(expansion bus interface)(214)는 직접 컴포넌트 연결(direct component connection)에 의하여 PCI 로컬 버스(206)에 연결된다. 대조적으로, 오디오 어댑터(216), 그래픽 어댑터(218) 그리고 오디오/비디오 어댑터(A/V)(219)는 확장 슬롯(expansion slot)으로 삽입되는 애드-인 보드에 의하여 PCI 로컬 버스(206)에 연결된다. 확장 버스 인터페이스(expansion bus interface)(214)는 키보드(keyboard) 및 마우스 어댑터(mouse adapter)(220), 모뎀(modem)(222), 그리고 추가의 메모리(224)를 위한 연결을 제공한다. SCSI 호스트 버스 어댑터(host bus adapter)(212)는 도시된 예에서 하드 디스크 드라이브(hard disk drive)(226), 테이프 드라이브(tape drive)(228), 그리고 CD-ROM(230)을 위한 연결을 제공한다. 전형적인 PCI 로컬 버스 실행은 3개 또는 4개의 PCI 확장 슬롯 또는 애드-인 코넥터(add-in connector)를 지원할 것이다. 도시된 예는 마더보드(motherboard) 및 3개의 확장 슬롯 상에서 4개의 로드(load)를 포함한다.

이 분야의 숙련된 자들은 도 2의 하드웨어가 구현함에 따라 변할 수 있음을 이해할 것이다. 예를 들면, 광 디스크 드라이브(optical disk drive) 등과 같은 다른 주변 장치가 도 2에서 도시된 하드웨어에 부가, 또는 대신하여 사용될 수 있다. NC로서 구현된 데이터 프로세싱 시스템(200)이 도 2에 예시된 것보다 적은 수의 컴포넌트들을 포함할 수도 있다. 예를 들면, 많은 NCs는 디스크가 없을(diskless) 수 있거나, 또는 하드 디스크 드라이브(226)와 같은 오직 하나의 저장 장치를 가질 수 있다. 또한 데이터 프로세싱 시스템(200)은 서버로서 작업을 수행할 수도 있다. 도시된 예가 본 발명에 대하여 구조적 제한을 의미하는 것은 아니다.

본 발명은 분산 데이터 프로세싱 시스템에서 NCs를 위한 부트 시간을 최소화하기 위한 방법 및 시스템을 제공한다. 도 3에 따르면, 네트워크 컴퓨터를 관리하는데 사용되는 컴포넌트의 블록 다이어그램이 본 발명의 바람직한 실시예에 따라 도시되어 있다. 도 3에서, 서버(300)는, 서버(300) 내에 위치하는 다수의 부트 이미지(302-308)로부터 원격 NC 부트 이미지(302)를 통신 링크(312)를 통해 NC(310)로 다운로드(download)시킨다. 그 다음에, NC(310)는 알려진 또는 선택된 상태에서 서버에 의해 하이버네이션하게(hibernate)되고 이미지(314)를 생산하는데, 이 이미지는 NC(310)가 재시작되어 언하이버네이션(unhibernation)을 할 때, NC(310)를 재시작하는데 사용될 수 있다. 하이버네이션은 NC와 같은 컴퓨터의 상태를 저장함을 포함한다. 하이버네이션은 특정의 실행이고 데이터는 하드 디스크 드라이브 또는 스태틱(static) RAM과 같은 저장 장치 상에 위치한 파일 또는 로우 I/O 포맷(raw I/O format) 안에 저장될 수 있다. 초기화 바로 다음에 NC(310)를 하이버네이션하는 대신에, 나중에 또는 선택된 애플리케이션 또는 옵션(option)이 시작된 후에 특정하게 선택된 상태에서 이미지(314)가 생성되고 저장될 수 있다. 구현함에 따라서 하이버네이션은 주기적으로 수행될 수 있다. 이 상태는, NC가 다수의 유저인 경우에, 선택된 애플리케이션이 시작된 후와 개인적 환경 설정(personal preference) 전의 것일 수 있다. 하이버네이션은 다양한 기능을 포함할 수 있다. 첫 번째는, 가령 서버에 의한 실수 또는 유저의 코멘드에 응답하는 실수로 인한 전력 사고 때문에 전력 공급이 중단될 때에 하드 디스크와 같은 보조 메모리 장치 안에 복구 목적을 위한 컴퓨터의 데이터가 저장되는, 긴급 자동 데이터 복구 기능이다. 그 후에 전

력이 컴퓨터에 다시 공급되면 그 데이터는 복구되고 또한 전력 중단 전의 상태로 컴퓨터는 회복된다. 두번 째는, 컴퓨터를 사용하는 동안 컴퓨터가 일정 시간 동안 사용되지 않는 경우에 있어서의 절전 기능인데, 컴퓨터의 데이터가 보조 메모리 장치에 저장된 후에 전력 공급은 자동적으로 중단되며, 그 후에 전력이 컴퓨터로 재공급되면 그 데이터는 회복되고 컴퓨터는 그 전의 상태로 복귀한다. 도시된 예는 전력 사고의 경우나 또는 유저가 NC를 재시작하거나 켜는 경우에 사용되는 제 1 기능에 사용될 때이다.

다음에 NC가 부팅될 때를 위하여 NC(310)의 이미지(314)가 디스크, 펌웨어(firmware), 또는 NC(310) 내에 위치한, NC(310)에 직접적으로 부착된(attached) 다른 어떤 매체 또는 저장장치에 저장된다. 일반적으로, NC(310)는 그것이 로컬 언하이버네이션을 수행하여야 하는지 또는 네트워크 메커니즘을 사용하여 부트 이미지를 다운로드 하여야 하는지를 결정하기 위해 서버에 조회할 것이다. 부트 이미지는 하이버네이션 이미지와는 대조적으로 NC로 송신될 오퍼레이팅 시스템 이미지인데, 이는 알려진 상태로 저장된 오퍼레이팅 시스템의 축소된 작업 셋트(set)이다. 도시된 예에서, NC(310)는, NC(310) 상에 지역적으로 저장된 이미지로부터 부트 또는 언하이버네이션할 것인가, 아니면 서버(300)로부터 부트 이미지를 다운로드 할 것인가를 결정하기 위해 서버(300) 상에 위치한 NC 파일 또는 레지스트리(registry)(316)를 검사한다. 이 검사는 NC(310)의 초기화 부트 프로그램의 일부분이다. NC 파일(316)은 NC(310)의 하이버네이션 이미지의 위치에 대한 정보를 포함할 것이다. 이 위치 이미지는 로컬 언하이버네이션을 수행하기 위해서 NC(310)에 의해 사용된다.

도시된 예에서, 이 정보가 서버(300)에서 손실되면, NC(310)는 정상 원격 부트를 수행할 것이고, 서버(300)로부터 부트 이미지 및 다른 정보를 다운로드 할 것이다. 대안적으로, NC(310)로 하여금 정상 원격 부트를 수행하도록 확실하게 명령하는 정보가 실재할 수 있다. 로컬 부트 이미지가 구식이 되거나(outdate) 서버가 재하이버네이션하기를 원하는 경우에 정상 부트가 전형적으로 수행된다. 덧붙여, 부트 이미지(302-308) 및 NC 파일(316) 둘 모두는 저장 유닛(즉, 도 1에서 저장 유닛(106))와 같은 분산 데이터 프로세싱 시스템 안의 어디에서나 저장될 수 있다. 이 프로세스를 통해 서버가 클라이언트 NCs의 클라이언트 관리를 유지하게 한다. 이에 더하여, 모든 이미지가 로컬이고 장치 상태(machine state)가 이미 하이버네이션 이미지에서 사전에 결정되어 있기 때문에(즉, 이미지에 있는 애플리케이션 및 오퍼레이팅 시스템이 이미 초기화되었음) 이미지의 로딩이 빨라질 수 있다. 구현함에 따라서, 증가 하이버네이션(incremental hibernation)이 발생할 수 있다. 로컬 NC 상의 이미지의 일부분이 구식이 되는 경우가 발생할 수 있지만, 대부분의 정보는 여전히 유효하다. 이 경우에, 서버는 로컬 부트 이미지 중 새로운 부분을 제외한 나머지는 계속 사용될 수 있음을 지시하는 메시지를 송신 할 수 있고, 그 새로운 부분은 메시지와 함께 송신되거나 메시지에 의해 참조된다.

도 4에 따르면, 이미지의 다이어그램이 본 발명의 바람직한 실시예에 따라 도시되어 있다. 이미지(400)는 NC를 재시작하거나 또는 언하이버네이션하기 위해 사용되는 이미지이다. 이미지(400)는 도시된 예에서 NC에 직접 접속되는 저장 장치로 NC 내에서 지역적으로 저장된다. 이미지(400)는 헤더 정보(header information)(402), 상태 정보(state information)(404), 페이징 파일 정보(paging file information)(406)를 포함한다. 헤더 정보(402)는 NC 상에서 하이버네이션 이미지를 위한 파일 레이아웃(file layout) 및 로딩 정보를 나타내고, 반면에 상태 정보(404)는 NC 내에서 물리적 메모리 및 하드웨어 장치(즉, 오퍼레이팅 시스템, 인터럽트(interrupt), 그리고 실시간 클락(real time clock))의 상태를 포함한다.

이미지(400)는 가상 메모리 관리를 포함하는 시스템만을 위한 페이징 파일 정보(406)를 포함할 것이다. 이미지(400) 내의 페이징 파일 정보는 페이징 파일 그 자신일 수 있다. 대안적으로, 페이징 파일의 일부분이 변하지 않도록 고정되거나 락킹될 수 있는데, 이렇게 고정된 부분은 NC를 회복하고, 부트하고, 언하이버네이션하는데 필요한 정보를 포함할 수 있다. 도시된 예가 페이징 파일을 사용하지만, 오퍼레이팅 시스템의 형식에 따라 페이징 파일 대신 페이징 스페이스(paging space)가 사용될 수도 있다. 페이징 스페이스 또는 페이징 파일은 가상 메모리 관리자를 가진 시스템에 적용 가

능하다.

도 5에 따르면, 본 발명의 바람직한 실시예에 따라서 하이버네이션을 위해 네트워크 컴퓨터를 구성(configuration) 하기 위한 프로세스의 플로우차트가 나타내어진다. 이 도면은 네트워크 컴퓨터의 로컬 부팅을 위해서 필요한 정보를 저장하기 위한, 원격 또는 네트워크 부트에서 사용되는 프로세스를 나타낸다. 프로세스는 전원이 NC에 의해 켜짐으로써 시작한다(스텝 500). 그 다음에, 데이터는 부트 디바이스로부터 획득된다(스텝 502). 부트 디바이스는 초기 프로그램 로드 정보를 저장하는 장치이고 예를 들면, 하드 드라이브 또는 펌웨어일 수 있다. 이 부트 디바이스가 로컬 부트 디바이스인가에 대해 결정이 이루어진다(스텝 504). 부트 디바이스가 로컬 디바이스이면, 로컬 I/O가 사용되고(스텝 506), 프로세스는 부트 디바이스로부터 오퍼레이팅 시스템을 로딩한다(스텝 508). 스텝(504)을 다시 참조하여, 부트 디바이스가 로컬이 아니면, 네트워크 컴퓨터를 초기화하기 위해 데이터를 획득하는데 있어서 네트워크 I/O가 사용되고(스텝 510), 전송된 바와 같이, 프로세스는 스텝(508)으로 진행한다. 그리고 나서, 오퍼레이팅 시스템은 완전히 로딩되고 초기화된다(스텝 512).

그 다음에, 유저 애플리케이션이 시작되고 초기화된다(스텝 514). 이것은 애플리케이션과 오퍼레이팅 시스템을 분리하는 상황을 배제하지 않는데, 그 분리에서, 둘 중 하나는 하이버네이션되는 중에도 나머지가 서버에 의해 전달될 수 있다는 뜻이다. 도시된 예에서, NC를 부팅 하는 것은 선택된 애플리케이션을 초기화함을 포함한다. NC가 부팅을 종료했을 때, 부트 디바이스가 로컬인가에 대해 결정이 이루어진다(스텝 516). 부트 디바이스가 로컬 디바이스가 아니라면, NC가 부팅 되었음을 지시하는 메시지를 NC가 서버로 송신한다(스텝 518). 로컬 부트의 경우에, 여전히 NC는 시스템이 재부팅 되었다는 지시를 서버에 송신한다. 그 다음에, NC는 하이버네이션을 위한 이미지를 하이버네이션할 것인가 또는 저장할 것인가의 메시지를 서버로부터 수신하고(스텝 520), 이미지를 로컬 미디어에 하이버네이션한 후(스텝 522), 부트 프로세스는 종료한다. 대안적으로, 구현하기에 따라서 이미지의 하이버네이션은 새로운 애플리케이션을 여는 것과 같은 로컬 유저 행동에 의하거나 또는 때때로 이미지를 주기적으로 하이버네이션시키는 로컬 타이머에 의하여 시작될 수도 있다. NC는 이미지의 하이버네이션이 발생한 후에 동작을 계속할 수 있다. 스텝(516)을 다시 참조하면, 부트 디바이스가 로컬인 경우 NC는 계속 작동하면서 사용자 입력을 처리하고 프로세스는 종료한다.

도 6에 따르면, 네트워크 컴퓨터를 부팅하기 위한 프로세스의 플로우차트가 본 발명의 바람직한 실시예에 따라 도시되어 있다. 프로세스는 네트워크 컴퓨터에서 전원을 켜는 것과 같은 시스템 이벤트(system event)로 시작된다(스텝 600). 그 다음에, 디바이스는 BIOS 에서 초기화되고(스텝 602), 제어는 오퍼레이팅 시스템으로 넘어간다(스텝 604). 그리고 나서, 네트워크 컴퓨터는 언하이버네이션할 것인가를 결정하기 위해 서버를 폴링(poll) 한다(스텝 606). 폴링은 알려진 네트워크 프로토콜(protocol)을 사용하여 실현될 수 있다. 언하이버네이션은, 예를 들면 디바이스 상태를 회복하는 것, 물리적 메모리를 로드 하는 것, 페이징 스페이스를 사전에 하이버네이션된 상태로 다시 셋트(set) 하는 것과 같은 물리적 상태로 시스템의 저장된 상태를 다시 회복하는 것을 의미한다. NC가 로컬 언하이버네이션을 수행하면, 서버는 하이버네이션된 이미지의 속성(property) 및 위치를 나타내는 정보를 넘겨준다(return, 스텝 608). 서버에 의하여 사용될 수 있는, 로컬 언하이버네이션을 수행하는지에 대한 결정 기준(criteria) 또는 서버로부터의 이미지를 사용하기 위한 결정 기준의 예들에는 하드웨어에 새로운 디바이스의 추가, 네트워크 컴퓨터에 의해 사용되는 애플리케이션에서의 변화 및 오퍼레이팅 시스템의 갱신이 포함된다. 그리고 나서, 네트워크 컴퓨터는 로컬 하이버네이션 이미지로부터 그 자신을 바라보는 상태로 복귀시키고(스텝 610), 부트 프로세스는 종료한다.

스텝(606)을 다시 참조하면, 서버가 정상 네트워크 부트가 채용되기를 지시한다면, NC를 원격 부팅을 수행하기 위해서 정상 네트워크 I/O가 발생하고(스텝 612), 그런 후에 부트 프로세스는 종료한다. 서버는, 예를 들면 갱신된 오퍼레이팅 시스템, 애플리케이션, 또는 구성과 같은 새로운 데이터를 다운로드 하기 위해 정상 네트워크 부트를 채용하도록 지시할 수 있다. 덧붙여, 새로운 애플리케이션이 서버로부터 NC로 로드될 수 있다. 정상 네트워크 부트 이후에, NC는

본 발명의 프로세스를 사용하여 하이버네이션을 위한 시스템의 이미지를 저장하도록 지시될 수 있다.

로컬 하이버네이션된 이미지는 네트워크로부터의 원격 부팅보다 빠른 NC 초기화를 제공한다. 이 프로세스는 NC가 선택된 상태로 회복되기 때문에 오퍼레이팅 시스템, 디바이스 드라이버, 그리고 애플리케이션을 로딩하고 초기화해야 하는 로컬 부팅보다 빠르다.

따라서, 본 발명은 네트워크 컴퓨터를 회복하기 위한 개선된 방법 및 장치를 제공한다. 이 장점은, NC 내의 하드웨어 상태, 물리적 메모리, 그리고 여러 가지 소프트웨어 컴포넌트를 포함하는 시스템의 '스냅샷'('snapshot')을 취함으로써 제공된다. 본 발명은 가상 페이지를 허용하는 장치를 위해서, 바라는 상태의 페이지 파일에 대한 복사본을 저장하거나 바라는 상태의 페이지 파일 중 일부분을 락킹할 수 있다. NC의 이 이미지 또는 '스냅샷'('snapshot')은 디스크, 펌웨어, 기타 NC에 대한 로컬 매체에 저장된다. 그리고 나서, NC는 원격 부팅에 의해서나 또는 스스로 로컬 초기화를 수행함으로써 그 자신을 초기화하는 방법이 아니라, 상기 이미지를 로딩함으로써 요구된 상태로 회복될 수 있거나 위치할 수 있다. 이 방법으로, NC가 분산 데이터 프로세싱 시스템 내에서 초기화되거나 또는 부팅될 때, 병목 현상(bottlenecks) 또는 '부트 스톰'('bootstorm')을 피할 수 있다. 또한, 본 발명은 서버로 하여금 여러 NC들의 클라이언트 관리를 유지하도록 허용한다.

본 발명이 완전히 기능적 데이터 프로세싱 시스템의 상황에서 기술되었지만, 이 분야의 당업자가 본 발명의 프로세스가 명령들의 컴퓨터 판독 가능 매체 형태 그리고 다양한 형태에서 배포될 수 있고, 본 발명이 배포를 수행하는데 실제로 사용되는 상호 수용 매체의 특별한 형식에 상관없이 동등하게 적용된다는 것을 이해하는 것은 매우 중요하다. 컴퓨터 판독 가능 매체의 예는 플로피 디스크, 하드 디스크 드라이브, RAM, CD-ROM과 같은 기록가능 - 형식 매체 그리고 디지털 및 아날로그 통신 링크와 같은 전송 - 형식 매체를 포함한다.

본 발명의 서술은 설명과 예시를 목적으로 제공되었으며, 본 발명이 개시된 형태로 제한되는 것은 아니다. 많은 변경과 변형이 이 분야의 당업자들에게는 당연한 것으로 받아들여질 것이다. 예를 들면, 본 발명의 프로세스는 네트워크 프로토콜에만 적용되는 것은 아니며, 많은 분산 데이터 프로세싱 시스템에 적용될 수 있다. 실시에는 본 발명의 사상을 가장 잘 설명하기 위해 선택되고 기술되었고, 당업자들이 고려되는 특별한 사용에 적합한 다양한 변경으로서의 다양한 실시예 형태로 본 발명을 이해할 수 있게 하는 실제적 응용을 가장 잘 설명하기 위해 선택되고 기술되었다.

발명의 효과

본 발명에 따르면, 개선된 분산 데이터 프로세싱 시스템(distributed data processing system)에서, 데이터 프로세싱 시스템을 관리하기 위한 개선된 방법 및 장치를 얻을 수 있으며, 특히 데이터 프로세싱 시스템을 부팅(booting)하기 위한 개선된 방법 및 장치를 얻을 수 있는 이점이 있다.

(57) 청구의 범위

청구항 1.

분산 데이터 프로세싱 시스템(distributed data processing system) 내의 데이터 프로세싱 시스템을 초기화하기 위한 방법에 있어서,

상기 데이터 프로세싱 시스템을 초기화하기 위해 사용되는 데이터를 원격 네트워크 소스(network source)로부터 상기 데이터 프로세싱 시스템에서 수신하는 단계와,

상기 데이터 프로세싱 시스템의 초기화 이후에 상기 데이터 프로세싱 시스템 내에서 상기 데이터 프로세싱 시스템의 이미지를 선택된 상태로 저장하는 단계 - 여기서 상기 이미지(image)가 저장된 이미지임 - 와,

상기 저장된 이미지를 사용하여 상기 데이터 프로세싱 시스템을 회복할 것인지를 결정하기 위해 상기 분산 데이터 프로세싱 시스템 내의 서버에 조회하는 단계와,

상기 데이터 프로세싱 시스템을 상기 선택된 상태로 회복시키기 위해서 상기 저장된 이미지를 사용해야 한다는 결정에 대한 응답으로 상기 저장된 이미지를 사용하여 상기 데이터 프로세싱 시스템을 상기 선택된 상태로 회복하는 단계

를 포함하는 것을 특징으로 하는 데이터 프로세싱 시스템을 초기화하는 방법.

청구항 2

제 1 항에 있어서, 상기 데이터 프로세싱 시스템에 의해 수신된 상기 데이터가 부트(boot) 이미지인 것을 특징으로 하는 데이터 프로세싱 시스템을 초기화하는 방법.

청구항 3.

제 1 항에 있어서, 상기 데이터 프로세싱 시스템이 물리적 메모리(physical memory)를 포함하고, 상기 저장된 이미지가 상기 물리적 메모리의 상태에 관한 정보를 포함하는 것을 특징으로 하는 데이터 프로세싱 시스템을 초기화하는 방법.

청구항 4

제 1 항에 있어서, 상기 분산 데이터 프로세싱 시스템이 서버(server)를 포함하고,

상기 저장된 이미지가 상기 데이터 프로세싱 시스템을 상기 선택된 상태로 회복시키기 위해 사용되지 않는다는 결정에 응답하여, 상기 서버로부터의 부트 이미지를 사용하여 상기 데이터 프로세싱 시스템을 초기화하는 단계

를 더 포함하는 것을 특징으로 하는 분산 데이터 프로세싱 시스템을 초기화하는 방법.

청구항 5.

제 1 항에 있어서, 상기 선택된 상태의 상기 데이터 프로세싱 시스템은 오퍼레이팅 시스템(operating system) 및 복수의 애플리케이션(application)을 포함하고, 그리고 상기 데이터 프로세싱 시스템의 메모리에 위치한 상기 오퍼레이팅 시스템 및 상기 복수의 애플리케이션이 상기 저장된 이미지 내에 저장되는 것을 특징으로 하는 데이터 프로세싱 시스템을 초기화하는 방법.

청구항 6.

제 5 항에 있어서, 가상 메모리가 실재한다면, 페이징 파일(paging file)이 상기 저장된 이미지 내에 저장되는 것을 특징으로 하는 데이터 프로세싱 시스템을 초기화하는 방법.

청구항 7.

분산 데이터 프로세싱 시스템 내에서 제 2 데이터 프로세싱 시스템에 연결된, 분산 데이터 프로세싱 시스템 내의 제 1 데이터 프로세싱 시스템을 초기화하기 위한 방법에 있어서,

상기 분산 데이터 프로세싱 시스템 내의 상기 제 2 데이터 프로세싱 시스템으로부터 수신된 데이터를 사용하여 상기 제 1 데이터 프로세싱 시스템을 초기화하는 단계와,

상기 제 1 데이터 프로세싱 시스템을 초기화한 후에 상기 제 1 데이터 프로세싱 시스템의 어떤 상태의 이미지를 저장하는 단계 - 여기에서, 저장된 이미지가 상기 제 1 데이터 프로세싱 시스템 내에 생성되고 저장됨 - 와,

상기 제 1 데이터 프로세싱 시스템을 상기 상태로 회복할 것인가를 결정하는 단계 및

상기 제 1 데이터 프로세싱 시스템을 상기 상태로 회복하라는 결정에 응답하여 상기 저장된 이미지를 사용하여 상기 제 1 데이터 프로세싱 시스템을 상기 상태로 회복하는 단계

를 포함하는 것을 특징으로 하는 제 1 데이터 프로세싱 시스템을 초기화하기 위한 방법.

청구항 8.

제 7 항에 있어서,

상기 저장된 이미지를 사용하여 상기 제 1 데이터 프로세싱 시스템을 회복하는 결정의 부재에 응답하여 상기 제 1 데이터 프로세싱 시스템에서 상기 제 2 데이터 프로세싱 시스템으로부터 데이터를 수신하는 단계

를 더 포함하는 것을 특징으로 하는 제 1 데이터 프로세싱 시스템을 초기화하기 위한 방법.

청구항 9.

분산 데이터 프로세싱 시스템에 있어서,

초기화 데이터를 포함하는 서버 컴퓨터; 및

복수의 클라이언트 컴퓨터(client computer) 내의 클라이언트 컴퓨터가 상기 서버 컴퓨터로부터 상기 초기화 데이터를 수신하고, 상기 초기화 데이터를 사용하여 선택된 상태로 초기화하고, 상기 클라이언트 컴퓨터 내에 선택된 상태의 이미지를 저장하고, 상기 클라이언트 컴퓨터를 시작하는 것에 대한 응답으로 상기 저장된 이미지를 사용할 것인지를 상기 서버 컴퓨터에 조회하고, 상기 저장된 이미지가 사용되어야 한다는 결정에 대한 응답으로 상기 이미지를 사용하여 상기 클라이언트 컴퓨터를 상기 선택된 상태로 회복하는, 복수의 클라이언트 컴퓨터

를 포함하는 것을 특징으로 하는 분산 데이터 프로세싱 시스템.

청구항 10.

데이터 프로세싱 시스템에 있어서,

데이터 프로세싱 시스템을 초기화하기 위해 사용되는 데이터를 상기 데이터 프로세싱 시스템으로 수신하기 위한 수신 수단과,

상기 데이터 프로세싱 시스템의 초기화 이후, 선택된 상태에서 상기 데이터 프로세싱 시스템 내에 상기 데이터 프로세싱 시스템의 이미지를 저장하기 위한 저장 수단 - 여기에서, 상기 이미지가 저장된 이미지임 - 과,

상기 저장된 이미지를 사용하여 상기 데이터 프로세싱 시스템을 회복할 것인지를 결정하기 위해서 상기 분산 데이터 프로세싱 시스템내의 서버에 대해 조회하는 조회 수단과,

상기 데이터 프로세싱 시스템을 상기 선택된 상태로 회복하기 위해서 상기 저장된 이미지가 사용되어야 한다는 결정에 대한 응답으로 상기 저장된 이미지를 사용하여 상기 데이터 프로세싱 시스템을 상기 상태로 회복시키기 위한 회복 수단

을 포함하는 것을 특징으로 하는 데이터 프로세싱 시스템.

청구항 11.

제 10 항에 있어서,

상기 데이터 프로세싱 시스템에 의해 수신된 상기 데이터가 부트 이미지인 것을 특징으로 하는 데이터 프로세싱 시스템.

청구항 12.

제 9 항에 있어서, 상기 데이터 프로세싱 시스템이 물리적 메모리를 포함하고, 상기 저장된 이미지가 상기 물리적 메모리의 상태에 관한 정보를 포함하는 것을 특징으로 하는 분산 데이터 프로세싱 시스템.

청구항 13.

제 9 항에 있어서,

상기 데이터 프로세싱 시스템이 서버에 연결되고

상기 저장된 이미지가 상기 데이터 프로세싱 시스템을 상기 선택된 상태로 회복시키기 위해 사용되지 않는다는 결정에 응답하여 상기 서버로부터 부트 이미지를 사용하여 상기 데이터 프로세싱 시스템을 초기화하기 위한 초기화 수단

을 더 포함하는 것을 특징으로 하는 분산 데이터 프로세싱 시스템.

청구항 14.

제 9 항에 있어서, 상기 상태에 있는 상기 데이터 프로세싱 시스템이 오퍼레이팅 시스템 및 복수의 애플리케이션을 포함하고, 상기 오퍼레이팅 시스템 및 상기 복수의 애플리케이션이 상기 저장된 이미지 내에 저장되는 것을 특징으로 하는 분산 데이터 프로세싱 시스템.

청구항 15.

제 14 항에 있어서, 페이징 파일이 상기 저장된 이미지 내에 저장되는 것을 특징으로 하는 분산 데이터 프로세싱 시스템.

청구항 16.

데이터 프로세싱 시스템을 초기화하기 위해서 데이터 프로세싱 시스템과 함께 사용하기 위한 컴퓨터 사용가능 매체 내의 컴퓨터 프로그램 제품에 있어서,

상기 데이터 프로세싱 시스템에서 상기 데이터 프로세싱 시스템을 초기화하기 위해 사용되는 데이터를 수신하기 위한 제 1 명령과,

상기 데이터 프로세싱 시스템의 초기화 이후 상태에서 상기 데이터 프로세싱 시스템 내에 상기 데이터 프로세싱 시스템의 이미지를 저장하기 위한 제 2 명령 - 여기에서 상기 이미지가 저장된 이미지임 - 과,

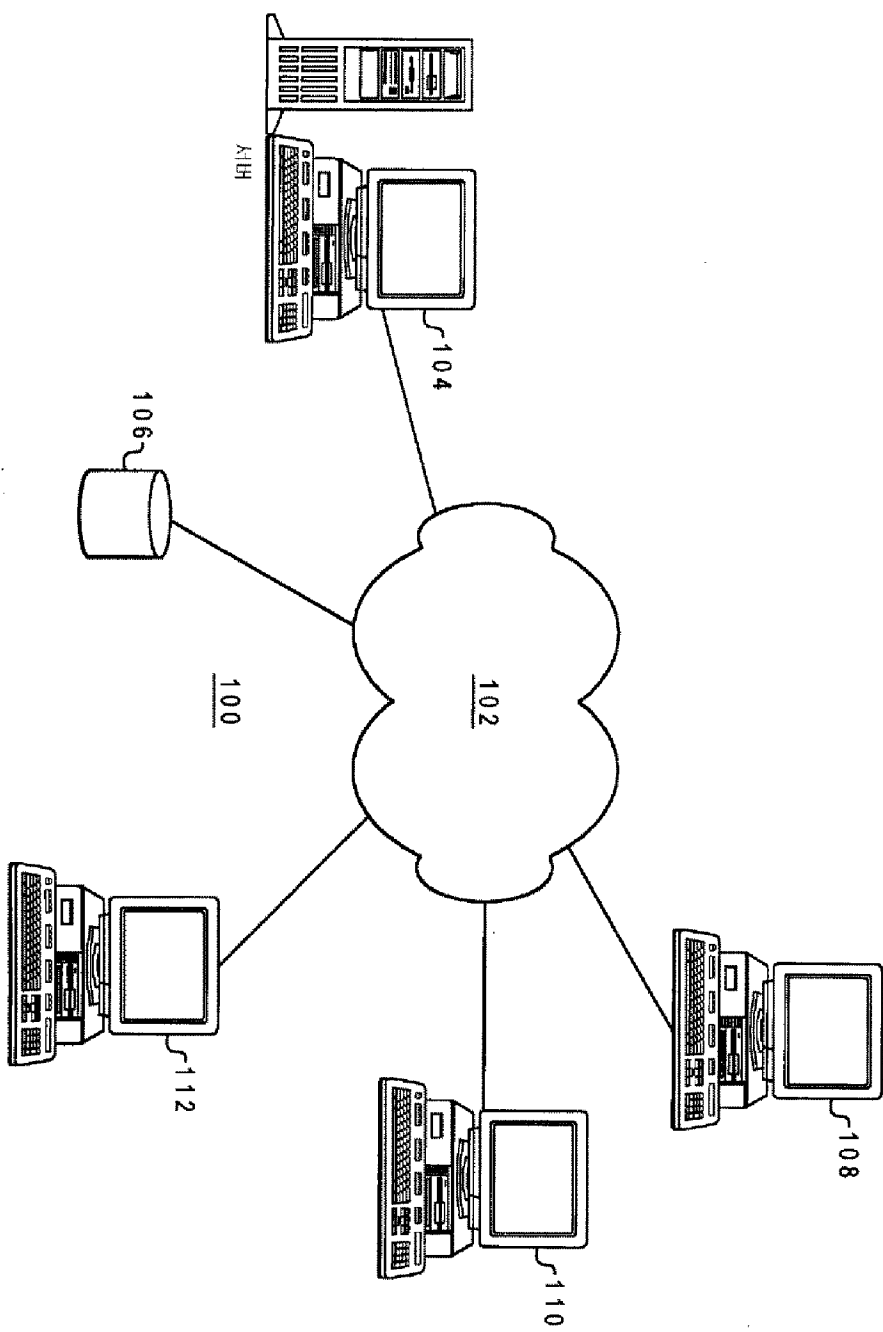
상기 저장된 이미지를 사용하여 상기 데이터 프로세싱 시스템을 회복시킬 것인지를 결정하기 위해 상기 분산 데이터 프로세싱 시스템의 서버에 조회하기 위한 제 3 명령과,

상기 데이터 프로세싱 시스템을 상기 선택된 상태로 회복하기 위해서 상기 저장된 이미지가 사용되어야 한다는 결정에 대한 응답으로 상기 데이터 프로세싱 시스템을 상기 저장된 이미지를 사용하여 상기 상태로 회복하기 위한 제 4 명령

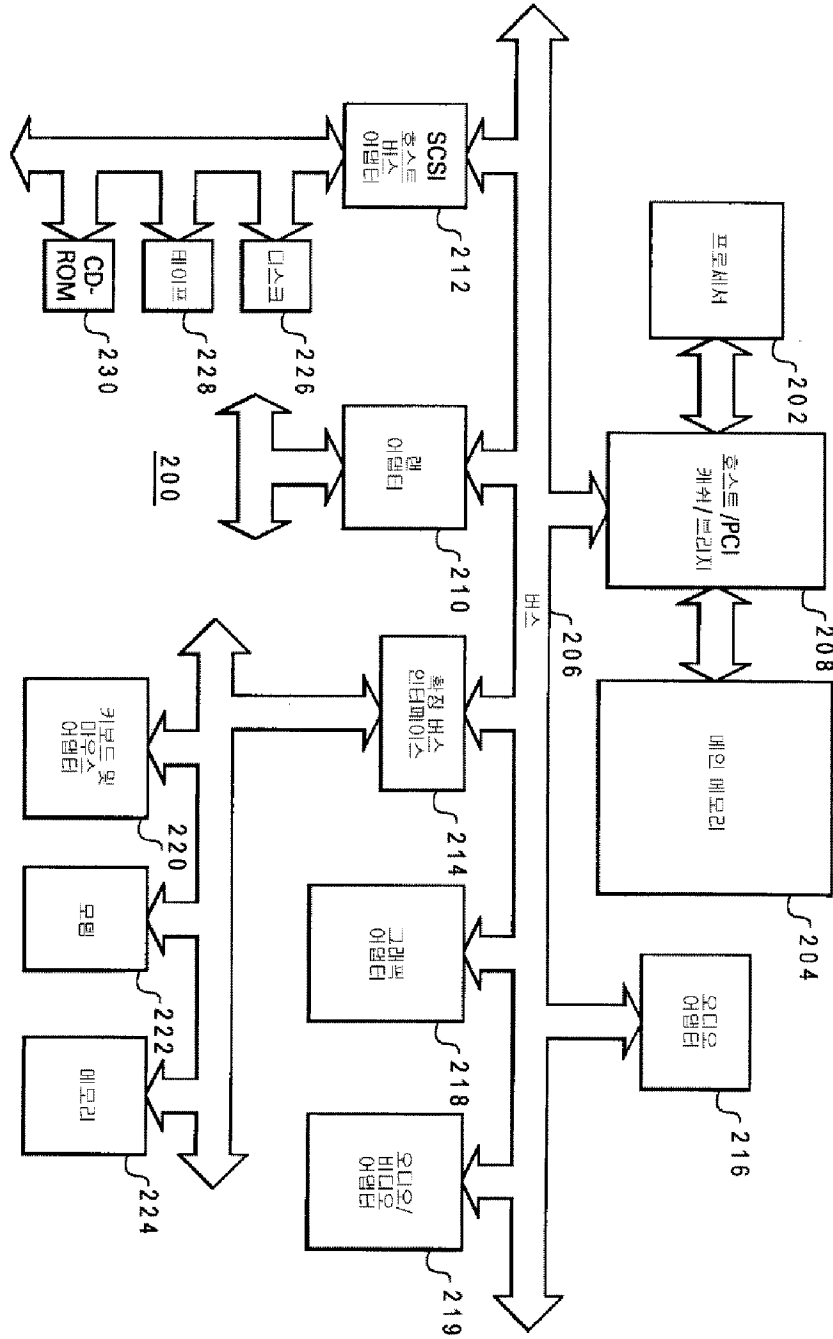
을 포함하는 것을 특징으로 하는 컴퓨터 프로그램 제품.

도면

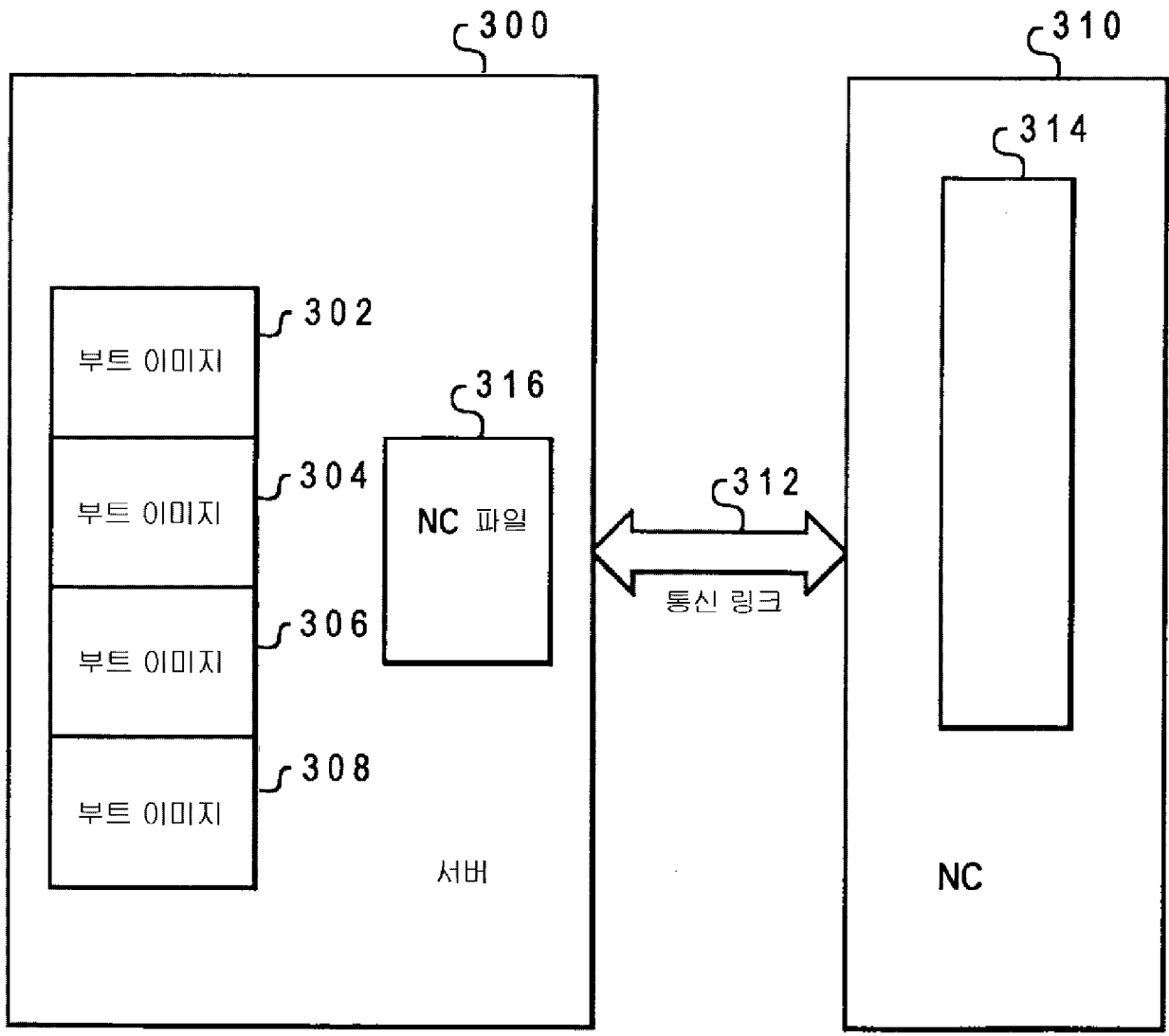
도면 1



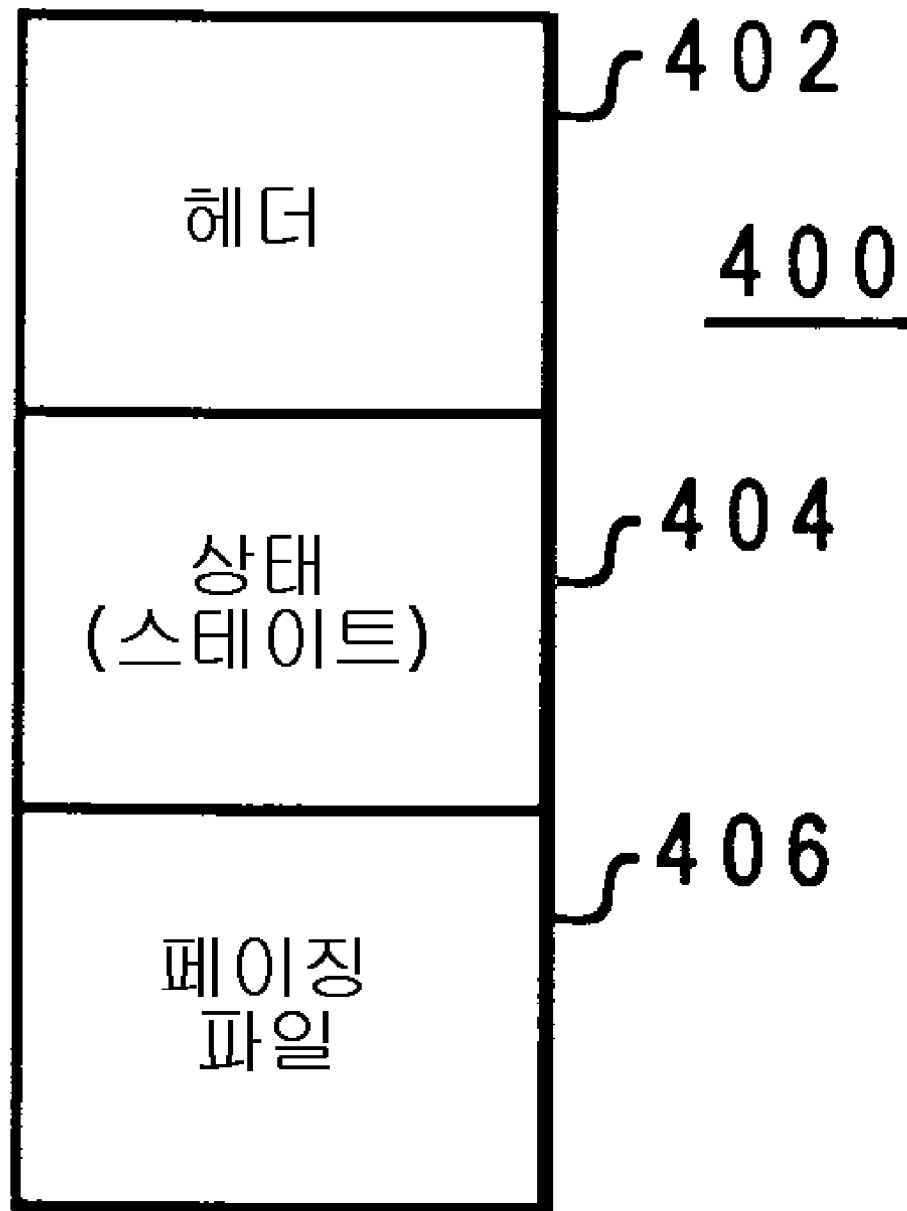
도면 2



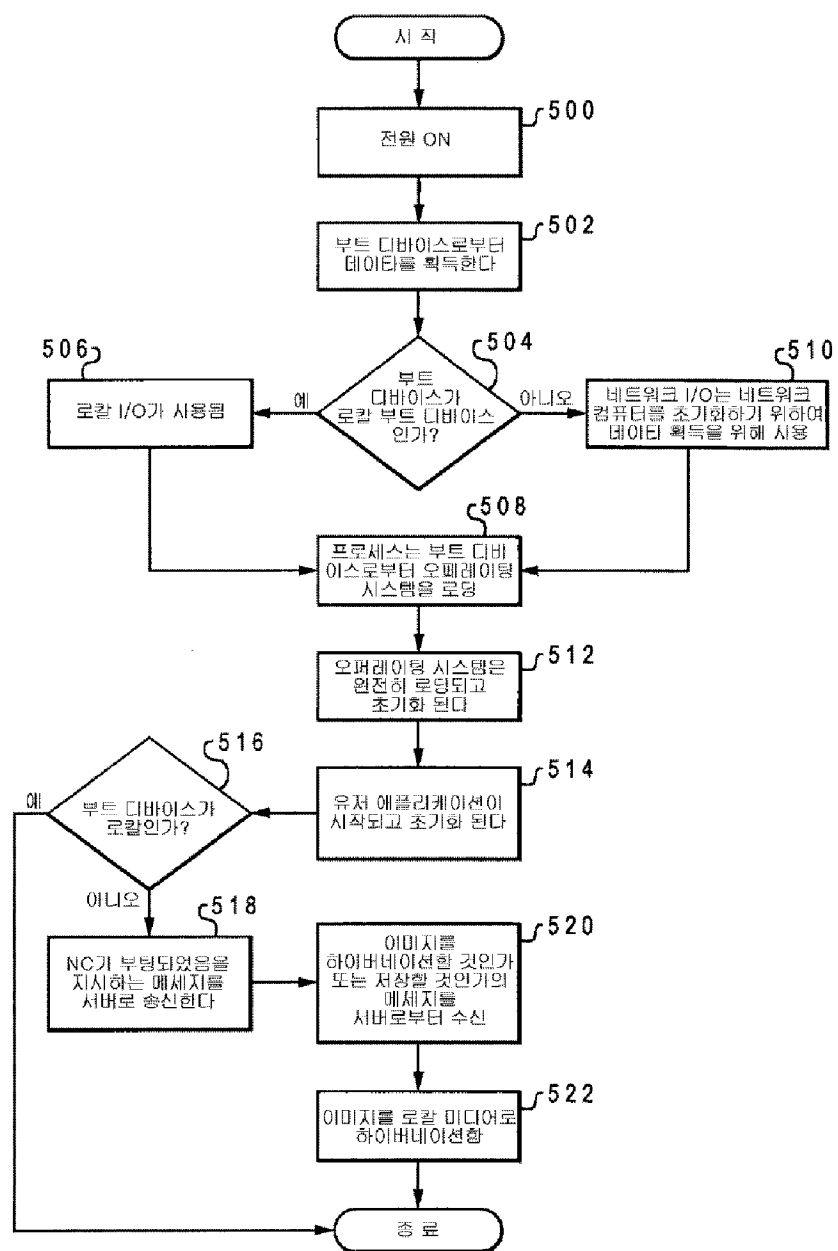
도면 3



도면 4



도면 5



도면 6

